

1

2

3

4

5

8

9

6

3

1 example, if a video channel is transmitted on a 100-Mbps isochronous
2 mode, a period of 40 microseconds is required during each 125-
3 microsecond cycle. Since the IEEE-1394 standard specifies that the
4 maximum amount of time available for isochronous transfer for each
5 cycle is 100 microseconds, the maximum number of video channels
6 which the current IEEE-1394 serial bus can support is only two.
7 Therefore, use of different speed nodes in a single IEEE-1394 serial bus
8 network represents a waste of otherwise usable bandwidth resource.

9 SUMMARY OF THE INVENTION

10 It is therefore an object of the present invention to reduce the
11 otherwise wasted bandwidth resource of an IEEE-1394 serial bus
12 network by providing a speed converter for converting the speed of
13 packets between nodes having different speed capabilities.

14 According to a first aspect of the present invention, there is
15 provided a speed converter for converting the speed of packets
16 transmitted between first and second communication nodes respectively
17 attached to first and second IEEE-1394 serial buses, comprising a first
18 transceiver node for receiving an inbound first packet at a first speed
19 from the first bus and transmitting an inbound second packet as an
20 outbound second packet at the first speed to the first bus, a second
21 transceiver node for transmitting the inbound first packet as an
22 outbound first packet at a second speed to the second bus and receiving
23 the inbound second packet at the second speed from the second bus, and
24 header translation circuitry for translating destination identifier of the
25 inbound first packet to destination identifier of the outbound first
26 packet according to a mapped correspondence between the first

0967150-092800

1 transceiver node and the second communication node, and translating
2 destination identifier of the inbound second packet to destination
3 identifier of the outbound second packet.

4 According to a second aspect, the present invention provides a
5 speed converter for converting the speed of packets transmitted
6 between a plurality of first communication nodes attached to a first
7 IEEE-1394 serial bus and a plurality of second communication nodes
8 attached to a second IEEE-1394 serial bus. The speed converter includes
9 at least one first repeater node connected to the first bus, a first
10 transceiver node for receiving an inbound first asynchronous packet
11 from the first bus at a first speed via the at least one first repeater node
12 and transmitting an inbound second asynchronous packet as an
13 outbound second asynchronous packet at the first speed to the first bus
14 via the at least one first repeater node, the first transceiver node having
15 identifiers identifying the first transceiver node itself and the at least
16 one first repeater node, at least one second repeater node connected to
17 the second bus, a second transceiver node for transmitting the inbound
18 first asynchronous packet as an outbound first asynchronous packet to
19 the second bus at a second speed via at least one second repeater node
20 and receiving the inbound second asynchronous packet from the second
21 bus at the second speed via the at least one second repeater node and
22 receiving the inbound second asynchronous packet at the second speed
23 from the second bus via the at least one second repeater node, the
24 second transceiver node having identifiers identifying the second
25 transceiver node itself and the at least one second repeater node, and
26 header translation circuitry for translating destination identifier of the

09671150-092800

11 According to a third aspect of the present invention, the speed
12 converter is provided for converting the speed of packets transmitted
13 between a plurality of first communication nodes attached respectively
14 to a plurality of first IEEE-1394 serial buses and at least one second
15 communication node attached to a second bus. The speed converter
16 comprises a plurality of speed conversion units associated respectively
17 with the plurality of first buses. Each speed conversion unit includes a
18 first transceiver node for receiving an inbound first packet at a first
19 speed from the associated first bus and transmitting an inbound second
20 packet as an outbound second packet at the first speed to the associated
21 first bus, a second transceiver node for transmitting the inbound first
22 packet as an outbound first packet at a second speed to the second bus
23 and receiving the inbound second packet at the second speed from the
24 second bus, and header translation circuitry for translating destination
25 identifier of the inbound first packet to destination identifier of the
26 outbound first packet according to mapped relationship between the

5 BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a speed converter according to a first embodiment of the present invention for converting the speed of packets transmitted on the IEEE-1394 serial bus;

Fig. 3 is a block diagram of a simplified IEEE-1394 serial bus network useful for describing the speed conversion of primary packets synchronized to cycle start packets;

Fig. 5 is a block diagram of a simplified IEEE-1394 serial bus network useful for describing mapping tables for mapping identifiers of the network nodes;

22 Figs. 6A and 6B show mapping tables associated with Fig. 5 for
23 translating source and destination identifiers of an inbound
24 asynchronous packet to source and destination identifiers of an
25 outbound asynchronous packet;

09130-0000

1 Fig. 7 is a sequence diagram illustrating an example asynchronous
2 data transfer using a write request packet and a write response packet;

3 Fig. 8A shows a data format of the stream control register provided
4 in each link layer processor;

5 Figs. 8B and 8C show example data formats of the stream control
6 registers provided respectively in link layer processors;

7 Fig. 9 shows data formats of master plug registers and plug control
8 registers of one of the link layer processors;

9 Fig. 10 is a flowchart illustrating a process for setting speed values
10 into the stream control registers of both link layer processors;

11 Fig. 11 is a block diagram of a speed converter according to a
12 second embodiment of the present invention;

13 Fig. 12 is a block diagram of a simplified IEEE-1394 serial bus
14 network associated with Fig. 11;

15 Figs. 13A and 13B show mapping tables associated with Fig. 12 for
16 translating source and destination identifiers of an inbound
17 asynchronous packet to source and destination identifiers of an
18 outbound asynchronous packet; and

19 Fig. 14 is a block diagram of a third embodiment of the present
20 invention.

21 DETAILED DESCRIPTION

22 In Fig. 1, a packet speed converter for an IEEE-1394 serial bus
23 network according to a first embodiment of the present invention is
24 designated by numeral 101. Speed converter 101 is a module that is
25 attached to IEEE-1394 serial buses B1 and B2 and includes a pair of
26 transceiver nodes 210 and 220. Although not shown in the drawings, an

0967150 "092800

Sub
B1

Transceiver node 210 includes a physical layer processor (LSI) 21 connected to the bus B1, a link layer processor (LSI) 31 and a speed setting switch 41 for setting a desired first speed value into the link layer processor 31. Likewise, the transceiver node 220 has a physical layer processor 22 connected to the bus B2, a link layer processor 32 and a speed setting switch 42 for setting a desired second speed value into the link layer processor 32. Physical layer processors 21, 22 and the link layer processors 31, 32 are designed to provide the functions specified by the IEEE-1394 Standard. Further, each physical layer processor and the associated link layer processor are interconnected via an interface specified by the IEEE-1394 Standard. Note that speed setting may also be achieved by coupling the speed setting switches 41 and 42 to the CPU 11 via the host bus S1 and setting desired speed values into the respective link layer processors 31, 32 from the CPU 11.

18 Link layer processors 31 and 32 are connected to a host bus S1 and
19 are interconnected by an isochronous data path S2 and a sync signal
20 path S3 for transmission of synchronized isochronous packets. Host bus
21 S1 serves as a data path for asynchronous packets.

Each of the speed setting switches 41, 42 has a plurality of speed setting values 0, 1, 2, 3, 4, 5 and 6 as shown in Fig. 2. Speed setting values 0, 1, 2, 3, 4, 5 and 6 correspond to respective speed conversion parameters. For speed setting values 0, 1, and 2, the speed of primary packets (either isochronous or asynchronous) is converted to 100 Mbps,

Shin⁸
By¹⁹

1 200 Mbps and 400 Mbps, respectively. For speed setting values 3, 4 and
2 5, the same conversion speed values are used for isochronous packets,
3 but the speed of asynchronous packets is converted to a maximum
4 possible value. For speed setting value 6, the speed converter performs
5 no speed conversion so that packets are transmitted at the same speed
6 as they are received. Link layer processors 31 and 32 transmit primary
7 packets according to the speed value set by the associated speed setting
8 switches 41, 42.

9 To perform speed conversion for asynchronous transfers, the speed
10 converter performs header translation. For this purpose, the speed
11 converter further includes a central processing unit 11, a read-only
12 memory 12 and a random access memory 13, all of which are connected
13 to the host bus S1. To achieve packet header translation, the CPU 11
14 executes programmed instructions of the present invention stored in
15 the read-only memory 12. As will be described, the RAM 13 maintains
16 mapping tables which define relationships between old and new
17 destination identifiers. An asynchronous packet transmitted at a first
18 speed from the bus B1, for example, is received by the transceiver node
19 210 and temporarily stored in the RAM 13. One of these mapping tables
20 is used by the CPU 11 for translating the source and destination
21 identifiers contained in the stored packet header to the identifiers of the
22 node 220 and a destination node attached to the bus B2. The header-
23 translated packet is then transmitted from the transceiver node 220 to
24 the bus B2 at a second speed.

25 In addition, for speed conversion of asynchronous packets, the CPU
26 11 has the function of segmenting a high speed packet into a series of

09671150-092800

1 low speed packets if the payload size of a transmitted high speed packet
2 exceeds the maximum payload size of the low speed packet, since the
3 maximum payload size of 400-Mbps packets is 2048 bytes while the
4 maximum payload size of 100 Mbps packets is 512 bytes.

5 On the other hand, channel number translation is performed for
6 isochronous (stream) transfers since the target node is identified by a
7 channel number instead of by a node identifier. For this reason, the
8 transceiver nodes 210 and 220 are respectively set to different channel
9 numbers before an isochronous transfer begins. As will be described in
10 detail later, stream packets transmitted on the bus B2, for example, are
11 received by the node 220 and passed through the isochronous data path
12 S3 to the node 210, where the channel number contained in their header
13 is translated to the channel number set in the node 210 and then
14 transmitted to the bus B1 at a second speed.

15 Note that the speed converter of this invention does not perform
16 transfer operations on all types of packets that propagate over the
17 associated buses. For example, all PHY packets transferred between
18 physical layers and all acknowledgment packets on the buses are not
19 transferred. The types of packets that are transferred through the speed
20 converter are the asynchronous packet and the stream packet which are
21 generally classified under the category of primary packets.

22 If two devices of different speeds are attached to the buses B1 and
23 B2 as represented by a 400-Mbps communication node 230 and a 100-
24 Mbps communication node 240 in Fig. 3, cycle start packets PS1 are sent
25 on bus B1 and cycle start packets PS2 are sent on bus B2, synchronized
26 to the cycle start packets PS1, as shown in Fig. 4. If a 400-Mbps primary

008260 051/960

1 packet PA1 is sent from node 230 on bus B1 immediately following a
2 cycle start packet PS1-1, the packet is translated to a 100-Mbps primary
3 packet PA2 and forwarded onto bus B2 immediately following a cycle
4 start packet PS2-2. Likewise, if a 100-Mbps primary packet PB1 is
5 transmitted from node 240 on bus B2 immediately following a cycle
6 start packet PS2-3, the packet is translated to a 400-Mbps primary
7 packet PB2 and forwarded onto bus B1 immediately following a cycle
8 start packet PS1-4. In this way, the speed converter allows other high
9 speed packets to be multiplexed on the bus B1 to achieve efficient
10 utilization of the bus B1. More specifically, since 100-Mbps standard
11 digital video signals take some 40 microseconds per cycle to travel over
12 the IEEE-1394 serial bus, the current system can support only two
13 channels for simultaneous transmission. Therefore, the speed converter
14 of this invention can support eight channels of 100-Mbps digital video
15 signals by converting their speed to 400 Mbps.

16 All nodes of the network are identified by a node identifier which
17 consists of a 16-bit bus ID and a physical ID. Since the speed converter
18 of this invention may be connected in an existing bus which is assigned
19 a single bus identifier, the buses B1 and B2 are assumed to be assigned
20 the same bus identifier, "3FF_h", for example. Thus, in the present
21 invention, the physical ID can be used to represent the node ID of each
22 node of the network. For asynchronous transactions, a packet from a
23 sending node contains its node ID in the source address field of its
24 header and the node ID of a destination node in the destination address
25 field.

008260"05112960

12722 12723 12724 12725 12726 12727 12728 12729 12730 12731 12732 12733 12734 12735 12736 12737 12738 12739 12740 12741 12742 12743 12744 12745 12746 12747 12748 12749 12750 12751 12752 12753 12754 12755 12756 12757 12758 12759 12760 12761 12762 12763 12764 12765 12766 12767 12768 12769 12770 12771 12772 12773 12774 12775 12776 12777 12778 12779 12780 12781 12782 12783 12784 12785 12786 12787 12788 12789 12790 12791 12792 12793 12794 12795 12796 12797 12798 12799 12800 12801 12802 12803 12804 12805 12806 12807 12808 12809 12810 12811 12812 12813 12814 12815 12816 12817 12818 12819 12820 12821 12822 12823 12824 12825 12826 12827 12828 12829 12830 12831 12832 12833 12834 12835 12836 12837 12838 12839 12840 12841 12842 12843 12844 12845 12846 12847 12848 12849 12850 12851 12852 12853 12854 12855 12856 12857 12858 12859 12860 12861 12862 12863 12864 12865 12866 12867 12868 12869 12870 12871 12872 12873 12874 12875 12876 12877 12878 12879 12880 12881 12882 12883 12884 12885 12886 12887 12888 12889 12890 12891 12892 12893 12894 12895 12896 12897 12898 12899 12900 12901 12902 12903 12904 12905 12906 12907 12908 12909 12910 12911 12912 12913 12914 12915 12916 12917 12918 12919 12920 12921 12922 12923 12924 12925 12926 12927 12928 12929 12930 12931 12932 12933 12934 12935 12936 12937 12938 12939 12940 12941 12942 12943 12944 12945 12946 12947 12948 12949 12950 12951 12952 12953 12954 12955 12956 12957 12958 12959 12960 12961 12962 12963 12964 12965 12966 12967 12968 12969 12970 12971 12972 12973 12974 12975 12976 12977 12978 12979 12980 12981 12982 12983 12984 12985 12986 12987 12988 12989 12990 12991 12992 12993 12994 12995 12996 12997 12998 12999 13000 13001 13002 13003 13004 13005 13006 13007 13008 13009 13010 13011 13012 13013 13014 13015 13016 13017 13018 13019 13020 13021 13022 13023 13024 13025 13026 13027 13028 13029 13030 13031 13032 13033 13034 13035 13036 13037 13038 13039 13040 13041 13042 13043 13044 13045 13046 13047 13048 13049 13050 13051 13052 13053 13054 13055 13056 13057 13058 13059 13060 13061 13062 13063 13064 13065 13066 13067 13068 13069 13070 13071 13072 13073 13074 13075 13076 13077 13078 13079 13080 13081 13082 13083 13084 13085 13086 13087 13088 13089 13090 13091 13092 13093 13094 13095 13096 13097 13098 13099 13100 13101 13102 13103 13104 13105 13106 13107 13108 13109 13110 13111 13112 13113 13114 13115 13116 13117 13118 13119 13120 13121 13122 13123 13124 13125 13126 13127 13128 13129 13130 13131 13132 13133 13134 13135 13136 13137 13138 13139 13140 13141 13142 13143 13144 13145 13146 13147 13148 13149 13150 13151 13152 13153 13154 13155 13156 13157 13158 13159 13160 13161 13162 13163 13164 13165 13166 13167 13168 13169 13170 13171 13172 13173 13174 13175 13176 13177 13178 13179 13180 13181 13182 13183 13184 13185 13186 13187 13188 13189 13190 13191 13192 13193 13194 13195 13196 13197 13198 13199 13200 13201 13202 13203 13204 13205 13206 13207 13208 13209 13210 13211 13212 13213 13214 13215 13216 13217 13218 13219 13220 13221 13222 13223 13224 13225 13226 13227 13228 13229 13230 13231 13232 13233 13234 13235 13236 13237 13238 13239 13240 13241 13242 13243 13244 13245 13246 13247 13248 13249 13250 13251 13252 13253 13254 13255 13256 13257 13258 13259 13260 13261 13262 13263 13264 13265 13266 13267 13268 13269 13270 13271 13272 13273 13274 13275 13276 13277 13278 13279 13280 13281 13282 13283 13284 13285 13286 13287 13288 13289 13290 13291 13292 13293 13294 13295 13296 13297 13298 13299 13300 13301 13302 13303 13304 13305 13306 13307 13308 13309 13310 13311 13312 13313 13314 13315 13316 13317 13318 13319 13320 13321 13322 13323 13324 13325 13326 13327 13328 13329 13330 13331 13332 13333 13334 13335 13336 13337 13338 13339 13340 13341 13342 13343 13344 13345 13346 13347 13348 13349 13350 13351 13352 13353 13354 13355 13356 13357 13358 13359 13360 13361 13362 13363 13364 13365 13366 13367 13368 13369 13370 13371 13372 13373 13374 13375 13376 13377 13378 13379 13380 13381 13382 13383 13384 13385 13386 13387 13388 13389 13390 13391 13392 13393 13394 13395 13396 13397 13398 13399 13400 13401 13402 13403

Sub
B.

When translating the header of an asynchronous packet

Fig. 7 is a sequence diagram of asynchronous transactions between communication nodes 231 and 241 when the speed setting switches 41

1 and 42 are respectively set to "2" (= 400 Mbps) and "0" (= 100 Mbps) (= 2 400 Mbps). At step SP1, the node 231 transmits a write request packet 3 at the set speed of 400 Mbps, with the source and destination fields 4 respectively set to the physical ID (= 2) of the source node 231 and 5 physical ID (= 3) of the transceiver node 210. Node 210 responds to the 6 write request packet with an ack_pending packet (step SP2). The write 7 request packet received by the node 210 is stored in the RAM 13. At step 8 SP3, the CPU 11 performs a header translation process by referencing 9 the mapping table 61 (Fig. 6A) to convert the source field of the packet 10 to the physical ID (= 1) of the transceiver node 220 on a predetermined 11 basis and convert the destination field to the physical ID (= 0) of node 12 241 according to the referenced mapping table 61. Subsequently, the 13 CPU 11 provides a header mapping process by identifying the write 14 transaction with a unique transaction label and mapping the old source 15 and destination IDs to the new source and destination IDs in the RAM 16 13. CPU 11 formulates a write request packet with a new header 17 containing the translated source and destination IDs and the transaction 18 label and forwards the packet to the link layer processor 32. Link layer 19 processor 32, knowing that the transmission speed is set equal to "0", 20 forwards the packet to the bus B2 at 100 Mbps (step SP4).

21 On receiving the write request packet from the bus B2, the 22 communication node 241 returns an ack_pending packet to the node 23 220 (step SP5). Then, the node 241 formulates a write response packet 24 with a header containing the transaction label and the source and 25 destination fields set to the physical ID (= 0) of its own node 241 and the 26 physical ID (= 1) of the transceiver node 220, respectively, and forwards

0067150.092800

4 The write response packet from node 241 is received by the node
5 220 and stored in the RAM 13. CPU11 examines the RAM 13 by
6 comparing the transaction label and the source and destination IDs
7 contained in the write response packet with those stored in the RAM 13,
8 and knows that node 220 has received a corresponding write response
9 packet from node 241 in response to the write request packet which the
0 node 210 had previously received from node 231.

11 A header translation process proceeds in the CPU 11 by replacing
12 the contents of the source and destination fields of the packet header
13 with the physical ID (= 3) of node 210 and the physical ID (= 2) of node
14 231, respectively. The header-translated write response packet is read
15 out of the RAM 13 and passed to the link layer processor 31 of node 210
16 (step SP8). Since the transmission speed is set equal to “3”, the link
17 layer processor 31 forwards the header-translated write response packet
18 to the bus B1 at 400 Mbps (step SP9). Node 231 receives this packet and
19 returns an ack_complete packet to the node 210 (step SP10).

20 If the communication node 241 initiates a transaction, the speed
21 converter 101 proceeds in the same manner as that described above
22 with the exception that the mapping table 62 (Fig. 6B) is used instead of
23 the mapping table 61.

Since most of low speed nodes currently available in the market issue transaction requests only to an isochronous resource manager that is attached to the same bus as the requesting node, the provision of only

Sub
BS

[illegible]

1 one mapping table may be sufficient for such nodes. In the above
2 example, the communication node 241 is the low speed node. If the
3 node 241 is of the type of node that issues transaction requests only to
4 the isochronous resource manager connected to the bus B2, the mapping
5 table 62 is not required.

6 For stream packet transfers, each of the link layer processors 31
7 and 32 has a 32-bit stream control register (SCR) whose format is shown
8 in Fig. 8A. The stream control register is divided into seven fields. The
9 first 2-bit field is a "send/receive" field which is used to indicate
10 whether a stream packet is to be transmitted to a bus or received from
11 the bus. Specifically, decimal "1" and "2" in the send/receive field
12 indicates reception and transmission, respectively. The second 6-bit
13 "channel" field is used to specify the channel number allocated by the
14 isochronous resource manager to the stream packet. A "1" or a "0" in
15 the one-bit "i" field respectively indicate that the stream packet is an
16 isochronous stream packet or an asynchronous stream packet. The 3-bit
17 "speed" field indicates the transmission speed of the stream packet,
18 with decimal "0", "1" and "2" respectively indicating the speeds of 100,
19 200 and 400 Mbps. The 4-bit "overhead" field and the 14-bit "payload"
20 field are used to specify the bandwidth necessary for the transmission of
21 the stream packet. The 2-bit "reserved" field is a field that is reserved
22 for future use.

23 If the transceiver node 210 transmits a 400-Mbps isochronous
24 stream packet to the bus B1 and the transceiver node 220 receives a 100-
25 Mbps isochronous stream packet from the bus B2, and channel numbers
26 "3" and "63" are assigned to the nodes 210 and 220, respectively, the

00671150.092800

According to the IEEE-1394 Standard, each node of the network has a configuration ROM in which the capability and functions of the node are stored. Assume that the communication node 231 initiates an isochronous transaction by transmitting a read request packet to the node 210 in the same manner as described above in connection with asynchronous transfers in order to know what functions the node 210 are capable of. In response to the read request packet from node 231, the transceiver node 210 accesses its own configuration ROM to read the functions of node 210. After header translation, the data read from the configuration ROM are set into the payload field of the read request packet and this header-translated packet is forwarded from node 220 to

1 node 241. In response, the node 241 accesses its own configuration ROM
2 to read its contents and returns a read response packet containing the
3 contents of the configuration ROM of node 241. After header
4 translation, the node 210 transmits the read response packet back to the
5 requesting node 231. Node 231 examines the contents of the read
6 response packet and determines the target node that can provide the
7 capability which is desired by the requesting node. Note that the
8 configuration ROM just described above is preferably in an address
9 space from "FFFF F000 0400" to "FFFF F000 07FC" defined on the
10 address space of each of the buses B1 and B2.

11 After determining the target node, different channel numbers are
12 set in the stream control registers of link layer processors 31 and 32
13 according to a flowchart shown in Fig. 10.

14 At step 301, the node 231 acquires a channel number (i.e., "3") from
15 the isochronous resource manager that is attached to the bus B1. Node
16 231 initiates a lock transaction to the node 210 by setting the acquired
17 channel number into the channel number field of the oPCR with and a
18 "1" into the point-to-point connection counter field of the oPCR (step
19 302). At step 303, the node 210 sets the send/receive field and channel
20 field of its own stream control register with values "2" and "3",
21 respectively. Thus, the node 210 is set in a transmit mode for
22 transmitting a stream packet of channel number "3" to the node 231. At
23 step 304, the node 220 sets a value of "1" in the send/receive field of its
24 own stream control register and a default value of "63" in its channel
25 field.

0067150.092800



Fig. 11 illustrates a second embodiment of the present invention. Speed converter 102 of this embodiment additionally includes physical layer processors 22 and 24 connected in series (daisy-chained) between the physical layer processor 21 and the bus B1, and a physical layer processor 25 connected in series between the physical layer processor 22 and the bus B2.

Since the physical layer processors 23, 24 and 25 all function as repeaters, they are designated as repeater nodes 212, 213 and 222, respectively. Similar to the previous embodiment, the link layer processor 31 and physical layer processor 21 function as a transceiver node 211 and the link layer processor 32 and physical layer processor 22 function as a transceiver node 221. Each of the transceiver nodes 211

1 and 221 further consists of a software-implemented transaction layer.

2 All nodes of the network are identified by a physical ID.

3 As shown in Fig. 12, communication nodes 311 and 312 are
4 attached to the bus B1 and communication nodes 321, 322 and 323 are
5 attached to the bus B2. For illustration, the nodes 311 and 321 are
6 assumed to be a digital video camera with a transmission speed of 200
7 Mbps, while the other nodes are personal computers capable of
8 operating at 400 Mbps.

9 Note that the link layer processor 31 of transceiver node 211 is
10 capable of receiving asynchronous packets from bus B1, containing not
11 only its own physical ID but also the physical IDs of the repeater nodes
12 212 and 213. Likewise, the link layer processor 32 of transceiver node
13 221 is capable of receiving asynchronous packets from bus B2,
14 containing the physical ID of the repeater node 222 as well as its own
15 physical ID. The speed of transmission of asynchronous packets from
16 each of the transceiver nodes 211 and 221 is set to the maximum
17 available value and the speed of transmission of stream packets is set to
18 400 Mbps.

19 Figs. 13A and 13B show two mapping tables 71 and 72 defined in
20 the RAM 13. In the mapping table 71, the transceiver node 211 and
21 repeater nodes 212, 213 are mapped to the communication nodes 321,
22 322 and 323 on the bus B2, respectively. In the mapping table 72, the
23 transceiver node 221 and repeater node 222 are mapped to the
24 communication nodes 311 and 312 on the bus B1, respectively.

25 If the transceiver node 211, for example, receives a configuration-
26 ROM read request packet from communication node 312 on bus B1.

008260 05117960

Sub
B1

1 Configuration ROM data of all communication nodes are stored in the
2 RAM 13. In response to the read request from node 312, the transceiver
3 node 211 reads from the RAM 13 the configuration ROM data of
4 communication node 321 on bus B2 that is defined in the mapping table
5 71 as a node corresponding to the transceiver node 211 and returns a
6 read response packet containing the read configuration ROM data.

7 Therefore, the communication nodes 321, 322, 323 on bus B2 are
8 "visible" from all communication nodes on bus B1, instead of the nodes
9 211, 212 and 213. Likewise, the communication nodes 311 and 312 on
10 bus B1 are "visible" from all communication nodes on bus B2, instead of
11 the nodes 221 and 222.

12 If the computer node 312 on bus B1 performs a configuration-ROM
13 read request transaction on the other nodes of bus B1, it will recognize
14 nodes 311 and 211 as a digital video camera. Likewise, if the computer
15 nodes 322 and 323 on bus B2 perform a configuration-ROM read request
16 transaction on the other nodes of bus B2, they will recognize nodes 321
17 and 221 as a digital video camera.

18 Following the configuration-ROM read request transaction, the
19 computer node 312 on bus B1 sends an asynchronous request packet to
20 the transceiver node 211. Specifically, the computer node 312 specifies
21 the allocated channel number and the set speed by performing a write
22 transaction on a register whose location is offset by "60C_h" from the
23 reference address value and starts a data transfer by performing a write
24 transaction on a register whose offset value is "614_h". Note that the
25 reference address value is written on the Unit Dependent Directory of
26 the configuration ROM of node 312.

Sub
B4
Cmt

09671150.092300

1 Transceiver node 211, on receiving the write request packet, stores
2 the packet in the RAM 13. CPU 11 performs a header translation by
3 rewriting the destination field of the request packet (which contains the
4 physical ID of node 211) with the physical ID of node 321 according to
5 the mapping table 71 and rewriting its source field (which contains the
6 physical ID of node 312) with the physical ID of node 222 according to
7 the mapping table 72.

8 The header-translated write request packet is then forwarded from
9 the RAM 13 to the transceiver node 221 and then transmitted at the
10 maximum speed of 200 Mbps to the digital video camera 321 via the
11 repeater node 222.

12 Digital video camera 321 responds to the write request packet with
13 a write response packet, which is received by the transceiver node 221
14 via the repeater node 222 and stored in the RAM 13 for header
15 translation. CPU 11 performs this header translation by rewriting the
16 destination field of the response packet (which contains the physical ID
17 of node 222) with the physical ID of node 312 according to the mapping
18 table 72 and rewriting its source field (which contains the physical ID of
19 node 321) with the physical ID of node 211 according to the mapping
20 table 71. The header-translated write response packet is forwarded
21 from the RAM 13 to the transceiver node 211 and then transmitted to
22 the computer node 312 at 400 Mbps which is the maximum transfer
23 speed between the nodes 211 and 312.

24 When the write transaction is successful, the speed converter 102
25 proceeds to set the stream control registers of the respective link layer
26 processors 31, 32 so that isochronous packets from the digital video

09671150.092800

1 camera 321 can be forwarded through the transceiver node 211 onto the
2 bus B1 at 400 Mbps.

3 In the IEEE-1394 serial bus network, bus reset is initiated under
4 various circumstances, which forces all nodes into their initialization
5 state, thereby initiating the configuration process. Preferably, the
6 transceiver node 221 is provided with a bus reset recovery feature to
7 minimize the interruption of data transfer caused by a bus reset. If bus
8 reset occurs on the bus B2 during the data transfer from the digital video
9 camera 321 to the transceiver node 221, the latter senses this condition
10 and performs a write transaction on the register of offset address value
11 "614_h" to enable the video camera 321 to reinitiate the isochronous
12 transfer by resetting the transmit/receive state of its stream control
13 register.

14 Prior to storage of the configuration ROM data of all
15 communication nodes of the network into the random access memory
16 13, the lower 64 bits of Bus_Info_Block of the configuration data and the
17 lower 64 bits of Node_Unique_Id leaf are preferably rewritten with the
18 EUI-64 values (Extended Unique Identifier, 64 bits) and the
19 module_vendor_id field of the Module_Vendor_Id entry is rewritten
20 with the company ID indicating the manufacturer of the speed
21 converter 102. The EUI-64, consisting of a 24-bit manufacturer's
22 identifier and a 40-bit chip identifier, is an identifier which is assigned
23 uniquely to all nodes of the network which are provided with the
24 general format configuration ROM.

25 When a configuration ROM read request packet is asserted on a
26 given transceiver node of speed converter 102, the rewritten

09671150-092300

1 configuration data is read from the RAM 13 and only the device
2 function is enabled to appear as if it were the same entity as the node
3 that corresponds in the mapping table to the given transceiver node. In
4 addition, the rewriting of the configuration ROM data eliminates the
5 need to alter the specifications of digital video controllers.

6 More specifically, if the transceiver node 211 receives a
7 configuration ROM read request packet from the bus B1, it reads
8 configuration ROM data from the memory 13 corresponding to the
9 destination identifier contained in the received read request packet and
10 transmits a read response packet to the bus B1 containing the read
11 configuration ROM data. If the transceiver node 221 receives a
12 configuration ROM read request packet from the bus B2, it reads
13 configuration ROM data from the memory 13 corresponding to the
14 destination identifier contained in the received read request packet and
15 transmits a read response packet to the second bus containing the read
16 configuration ROM data.

17 A third embodiment of the present invention is shown in Fig. 14 in
18 which speed converters 101A, 101B and 101C of the same configuration
19 as that of Fig. 1 are incorporated in a single speed converter 103. These
20 speed converters are set at different speed values. The transceiver nodes
21 210 of the speed converters 101A, 101B, 101C are connected to buses B1-
22 1, B1-2 and B1-3, respectively, and the transceiver nodes 220 of these
23 speed converters are connected in series to the bus B2. High speed
24 communication nodes 401, 402 and 403 are attached to the buses B1-1,
25 B1-2 and B1-3, respectively, and low speed communication nodes 404
26 and 405 are attached to the bus B2. Similar to the first embodiment, all

00671150.092800

1 the transceiver nodes 220 are recognized by the communication nodes
2 404 and 405 as if they were the high speed communication nodes 401,
3 402 and 403. In this way, data transfer can be provided at a number of
4 different speeds.

5 As described above, in the speed converter for an IEEE-1394 serial
6 bus network, a first transceiver node represents a low speed
7 communication node on one bus and performs data transfer with a high
8 speed communication node on the other bus and a second transceiver
9 represents the high speed communication node and performs data
10 transfer with the low speed communication node. A high speed device
11 can maintain its transmission speed when communicating with a low
12 speed device through the speed converter of the present invention, and
13 a substantial resource saving is achieved for the IEEE-1394 serial bus.
14 Experiments showed that more than three digital video channels were
15 successfully transmitted on the same IEEE-1394 serial bus.

008250" 05T 4960